Notice v. 1.0
Février 2008

Guide d'utilisation

de

ESA PetriNet

- Temporal Edition -

Version 1.0 (2007)

Copyright © LAAS-CNRS
Groupe ISI

TABLE DES MATIERES

1. Introduction 1

1.1 Description rapide du logiciel 1

1.2 Configuration requise 1

1.3 Description de la finalité du manuel 1

2. Présentation générale 2

2.1 Interface principal 2

2.2 Barre de menu 3

2.3 Fichier de structure d'un RdP 3

2.4 Fichier de communication 5

2.5 Editeur de fichier de communication 6

2.6 Fichier résultat 7

2.7 Fichier contraintes 8

2.8 Zone des graphes de précédences 8

2.9 Fichiers projet 9

2.10 Les messages d'erreurs 10

3. Les étapes de l'utilisation sur un exemple 11

3.1 Présentation de l'exemple 11

3.2 Editer les RdP avec TINA 12

3.3 Enregistrer les fichiers.ndr 13

3.4 Générer les fichiers-struct.txt 13

3.5 Compléter l'interface 13

3.6 Créer le fichier-com.txt 14

3.7 Générer les scenarios 15

3.8 Commenter les scenarios 16

3.9 Sauvegarder le projet 16

4. Note 17

1. INTRODUCTION

1.1 DESCRIPTION RAPIDE DU LOGICIEL

ESA PetriNet[1] est un logiciel développé par le groupe ISI[2] du LAAS[3]-CNRS. Il permet de générer les ensembles d'évènements qui conduisent un système dans un état particulier prédéfini. Ces ensembles d'évènements forment des scénarios, qui, dans le cas où l'état particulier représente un état de défaillance du système, sont appelés **scénarios redoutés**.

Par construction de son algorithme, ESA PetriNet fournit tous les scénarios strictement nécessaire et suffisant pour atteindre l'état spécifié (notions de **complétude** et de **minimalité**). Pour fonctionner, le logiciel a besoin d'une modélisation faite : soit d'un unique réseau de Petri, soit de plusieurs réseaux de Petri liés entre eux par des appels à méthodes, comme le permet l'aspect **orienté objets**.

1.2 CONFIGURATION REQUISE

ESA PetriNet est écrit en langage Java, c'est pourquoi il nécessite une machine virtuelle Java pour s'exécuter. La JDK[4] installée doit être la **JDK version 6** ou une version supérieure, puisque la généricité et certaines structures de données « dynamiques » ou non-contraintes (tel que *arraylist<E>* ou *hashmap<K,E>*) sont nécessaires.

Les réseaux de Petri peuvent être conçus à l'aide de l'outil TINA[5], développé par le groupe OLC[6] du LAAS-CNRS. Dans ce cas, la **version 2.8.4 de TINA** est recommandée. La compatibilité des versions antérieurs n'a pas été vérifiée (il faut au moins disposer des étiquettes (ou *label*) associées aux places et aux transitions du réseau de Petri) et la compatibilité des versions suivantes n'est pas garantie.

1.3 DESCRIPTION DE LA FINALITE DU MANUEL

Ce manuel a été rédigé pour des personnes ayant déjà des connaissances concernant les réseaux de Petri et leur concept d'invariant de marquage, la modélisation Orienté Objets de ces réseaux de Petri et toutes les notions relatives à celle de scénario.

Mise à part l'introduction, ce manuel s'articule en 2 grandes parties :

I. La première, intitulé « Présentation Générale », fournie toute l'information nécessaires à l'utilisateur qui souhaite tester et découvrir par lui-même le logiciel. Mais le principe étant celui de l'**accès direct à une section**, cette partie conviendra plus particulièrement aux personnes ayant déjà utilisé ESA PetriNet au moins une fois.

II. La deuxième partie, quant à elle, invite le lecteur à suivre un **tutorial**, d'où son titre : « les étapes de l'utilisation sur un exemple ». Le principe ici est donc plutôt celui de la **lecture continue**.

[1] Extraction Scenarios & Analyzer by Petri Net model
[2] Ingénierie Système et Intégration
[3] Laboratoire d'Analyse et d'Architecture des Systèmes
[4] Java Developement Kit
[5] TIme petri Net Analyzer
[6] Outils Logiciels pour la Communication

2. PRESENTATION GENERALE

2.1 INTERFACE PRINCIPAL

L'interface principal de ESA PetriNet (voir figure 1) est composée d'une barre de menu assez classique suivie par une barre d'outil (zone 1), d'un récapitulatif des fichiers du modèle (zone 2) auquel est adjoint une série de boutons permettant la gestion de ces fichiers (à droite de la zone 2), d'un ensemble de boutons principaux permettant de générer les scénarios ou les graphes de précédence (zone 3), d'un cadre principal réservé à l'affichage de ces graphes de précédence (zone 4) et enfin d'une barre d'état indiquant certaines actions réalisées ou des messages d'erreurs (zone 5) (voir la section 2.9 de ce manuel pour les explications des différents messages d'erreurs possibles).

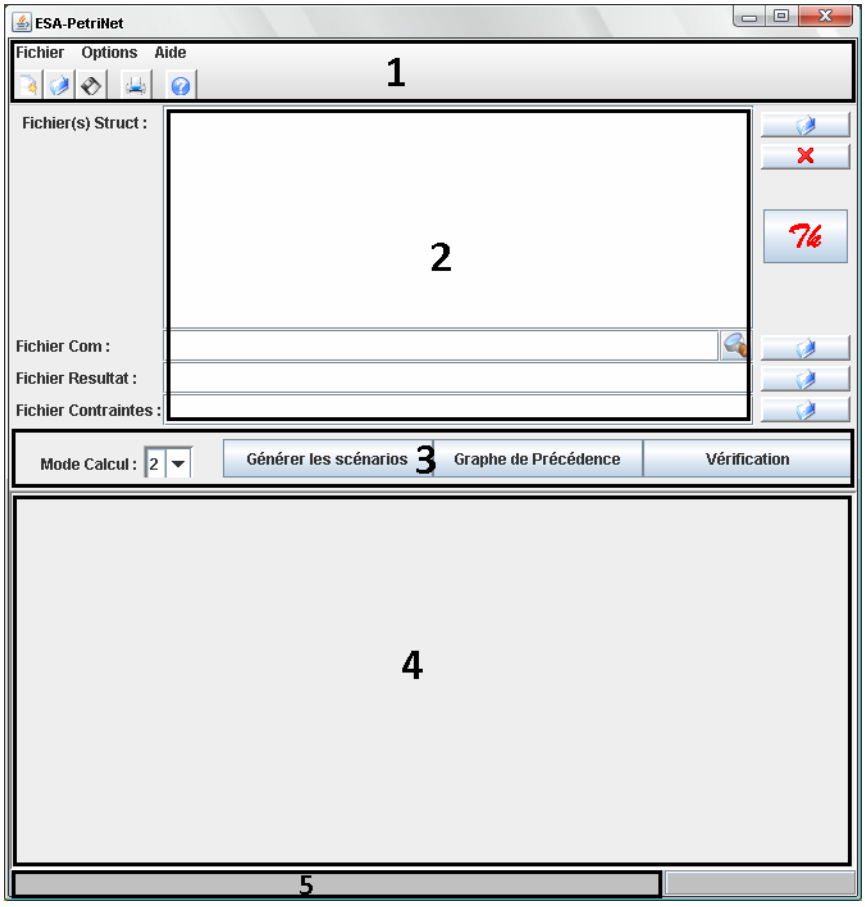


Figure 1 : Interface principal de ESA PetriNet

Le bouton en forme de loupe situé dans la zone 2 ouvre un éditeur de fichiers de communications, qui permet aussi bien de créer un nouveau fichier que de modifier un fichier existant (voir la section 2.4 pour plus de détails à propos des fichiers de communications et de leur éditeur).

L'outil TINA peut être appelé à l'aide du bouton rouge 74 à droite de la zone 2. Si aucun fichier de structure d'un RdP[7] n'est sélectionné, TINA s'ouvre avec un fichier vierge. Si un fichier de structure

[7] Réseau de Petri

d'un RdP est sélectionné **et si son fichier*.ndr* se trouve dans le même répertoire que ce fichier-*struct.txt***, TINA ouvrira directement le RdP sélectionné. Des modifications peuvent alors être faites sur le RdP, mais il ne faudra pas oublier de **procéder à une nouvelle analyse structurelle** du réseau pour obtenir le fichier de structure du RdP à jour (voir la section 2.3 pour plus de détails concernant les fichiers de structure d'un RdP).

2.2 BARRE DE MENU

La barre de menu contient 3 sous-menus : ***Fichier***, ***Options*** et ***Aide*** (voir figure 2).

Figure 2 : Barre de menu

Le sous-menu ***Fichier*** propose des commandes classiques tel que : créer un ***nouveau*** projet, ***ouvrir*** un projet existant, ***enregistrer*** le projet courant, ***imprimer*** la page (seul les graphes de précédences s'impriment) et ***quitter*** le logiciel.

Le sous-menu ***Options*** permet de spécifier l'***adresse de TINA***, c'est-à-dire le chemin de l'exécutable **nd.exe** sous Windows®, sans quoi il serait impossible d'appeler l'outil TINA depuis ESA PetriNet par l'intermédiaire du bouton prévu à cet effet (voir la section 2.1). Ce même sous-menu permet aussi de choisir entre la génération : de ***tous les scénarios***, des ***scénarios redoutés non-minimaux*** seulement ou des ***scénarios redoutés minimaux*** uniquement.

Le troisième et dernier sous-menu est celui intitulé ***Aide***. Comme son nom l'indique, il propose d'ouvrir l'***aide*** de l'outil, mais permet aussi d'afficher une fenêtre ***à propos*** (***about*** en anglais).

2.3 FICHIER DE STRUCTURE D'UN RDP

Les fichiers représentatifs des RdP utilisés par ESA PetriNet sont les fichiers-*struct.txt* générés par le module d'**analyse structurelle** de l'outil TINA, car ceux-ci contiennent, en plus des informations structurelles telles que les invariants de marquages, la structure du RdP sous forme textuelle.

Ainsi, les fichiers*.ndr* d'édition graphique ne sont pas utilisés par ESA PetriNet. La présence de ces fichiers*.ndr* n'est donc pas obligatoire, mais, placés dans le même répertoire que leurs fichiers-*struct.txt*, elle permet leurs ouvertures directes dans TINA depuis ESA PetriNet (voir section 2.1).

LE FORMAT DES ETIQUETTES ASSOCIEES AUX PLACES

Avant d'effectuer l'analyse structurelle du RdP, il faut bien « formater » les étiquettes (label) associées à chaque place du RdP. En effet, celles-ci sont utilisées pour définir le type de place et le

marquage initial du RdP global dans les conditions de fonctionnement normal *(attention : il ne s'agit pas ici du marquage utilisé pour débuter le raisonnement arrière)*. Les différents types de places possibles sont les suivants :

- « **N** » : places de fonctionnement normal,
- « **D** » : places appartenant à l'état partiel redouté à analyser,
- « **IR** » : places appartenant à un état défaillant réparable (mais non à l'état redouté),
- « **I** » : places appartenant à un état défaillent non-réparable (mais non à l'état redouté).

Le format utilisé pour les étiquettes des places est alors :

« ***Identificateur_type_de_place,nombre_de_jetons_dans_la_place*** »

Exemples : « **N,1** », « **D,0** », « **IR,0** », « **I,0** » …

LES INTERVALLES DE TIR DES TRANSITIONS

Il faut aussi définir les intervalles de tir des transitions **avant** de procéder à l'analyse structurelle pour que ces informations soient contenues dans les fichiers résultant de cette analyse. Dans cette première version d'ESA PetriNet, seuls les intervalles du type [a, ∞ [ou [a, b] sont acceptés (sachant qu'un intervalle non-spécifié est autorisé, puisqu'il signifie [0, ∞ [, ce qui est compris dans [a, ∞ [). Ainsi, il n'est pas possible d'utiliser les formes]a, b],]a, b[ou [a, b[.

LE MARQUAGE INITIAL

Le « véritable » marquage initial du ou des RdP, c'est-à-dire celui qui est effectivement représenté par la présence de jetons dans certaines places et non à travers l'information fournie par les étiquettes des places *(qui indiquent le marquage initial du RdP global dans les conditions de fonctionnement normal – voir quelques lignes plus haut)*, correspond à l'état partiel redouté à analyser.

LE FICHIER DE STRUCTURE

La figure 4 donne le fichier de structure du RdP « bien formaté » de la figure 3.

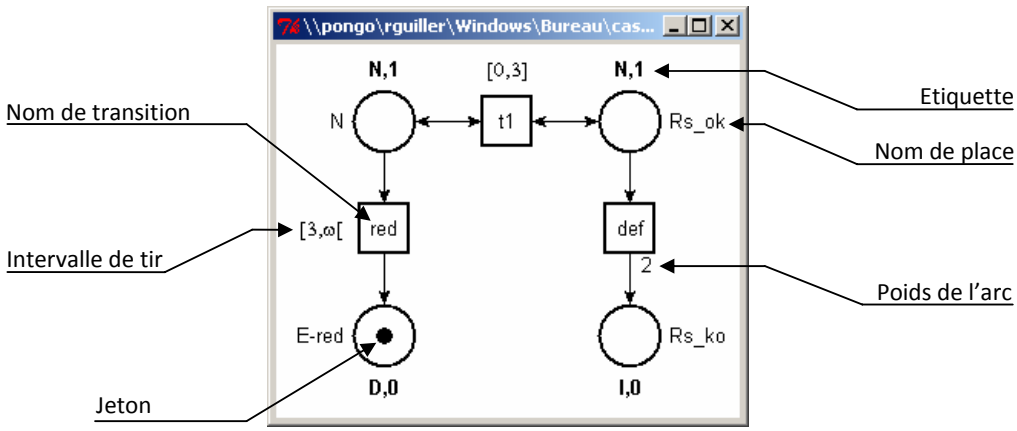


Figure 3 : Exemple de RdP « bien formaté »

```
Struct version 2.8.4 -- 10/27/06 -- LAAS/CNRS

parsed net cas_etude

4 places, 3 transitions

net cas_etude
tr def Rs_ok -> Rs_ko*2
tr red [3,w[ N -> {E-red}
tr t1 [0,3] N Rs_ok -> N Rs_ok
pl {E-red} : {D,0} (1)
pl N : {N,1}
pl Rs_ko : {I,0}
pl Rs_ok : {N,1}

0.000s

P-SEMI-FLOWS GENERATING SET ----------------------------------

invariant

Rs_ko Rs_ok*2
{E-red} N

0.000s

T-SEMI-FLOWS GENERATING SET ----------------------------------

not consistent

t1

0.000s

ANALYSIS COMPLETED  ------------------------------------------
```

Figure 4 : Exemple de fichier de structure

Il est a noté que seul les parties encadrées sur la figure 4 sont utilisées par ESA PetriNet – le reste peut éventuellement être supprimé.

Attention : évitez de nommer des transitions par : **i1**, **i2**,… , **e1**, **e2**,… ou **f1**, **f2**,… , car ces notations sont utilisées respectivement pour les événements d'**initialisation**, les événements d'**enrichissement** et les événements **finaux**.

2.4 FICHIER DE COMMUNICATION

Le fichier de communication est utile pour une modélisation orientée objets, c'est-à-dire lorsque plusieurs RdP interviennent. Mais il est parfaitement possible de n'utiliser qu'un seul RdP pour la modélisation : dans ce cas aucun fichier de communication ne doit être défini.

La structure de ce fichier est donnée en figure 5, où la transition t0 de l'objet 1 appelle les deux transitions nommées t1 des objets 2 et 3, et la transition t2 de l'objet3 appelle la transition t4 de l'objet4.

```
ESA-PetriNet version 1.0 - Fichier Communications -

objet1/t0 : objet2/t1 objet3/t1
objet3/t2 : objet4/t4
```

Figure 5 : Structure d'un fichier de communication (-com.txt)*

On constate qu'il est possible que 2 transitions de 2 objets différents aient le même nom, même si cela n'est pas recommandé afin d'éviter des confusions.

Note : il n'est pas possible de définir une transition qui appelle une ou des transitions et en même temps est appelée par une ou des transitions.

2.5 EDITEUR DE FICHIER DE COMMUNICATION

Le fichier de communication peut être créé à l'aide de l'éditeur de fichier Com fourni avec ESA PetriNet (voir figure 6) en cliquant sur le bouton une fois que la liste des fichiers de structure des RdP qui interviennent dans le modèle a été renseignée.

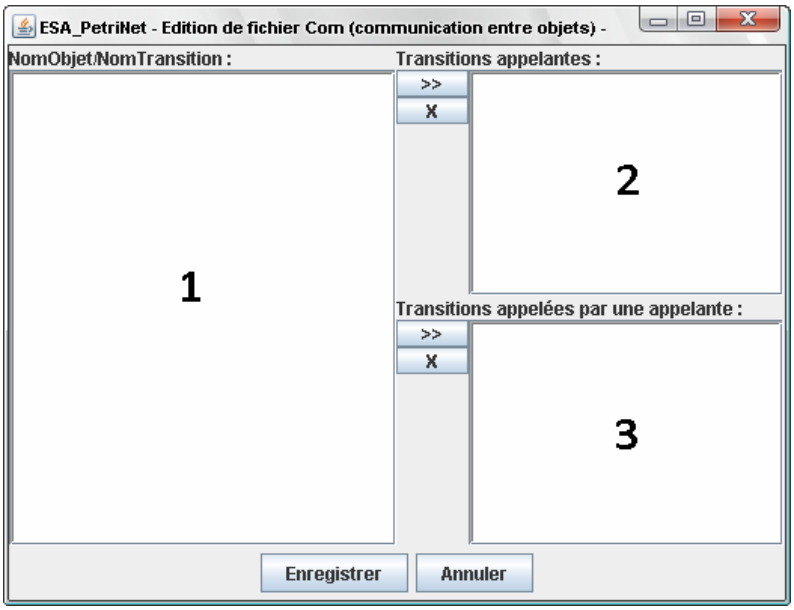


Figure 6 : Editeur de fichier de communication

Toutes les transitions du modèle apparaissent à gauche dans l'éditeur (zone 1). Le fonctionnement de l'éditeur est alors le suivant :

- La première étape consiste à définir les transitions appelantes, c'est-à-dire celles qui appellent une ou plusieurs autres transitions. Pour ce faire, il suffit de sélectionner la transition dans la liste de gauche (zone 1) et de l'ajouter dans la liste des transitions appelantes (zone 2) en cliquant sur le bouton « **>>** » à gauche de la zone 2.

- La deuxième étape consiste à définir les transitions appelées pour chaque transition appelante. Le principe est de sélectionner la transition appelante considérée (zone 2), puis de sélectionner la transition à appeler dans la zone 1, et enfin d'ajouter cette dernière transition à la liste des transitions appelées de la transition appelante considérée dans la zone 3 en cliquant sur le bouton « **>>** ».

Les 2 boutons « **X** », à gauche de la zone 2 et à gauche de la zone 3, permettent respectivement de supprimer la transition appelante sélectionnée ou la transition appelée sélectionnée.

2.6 FICHIER RESULTAT

Obtenu après la recherche de scénarios, le fichier-*result.txt* contient justement les scénarios générés. Il sert ensuite comme donnée d'entrée à la construction et l'affichage des graphes de précédence, forme beaucoup plus commode et lisible que l'écriture textuelle des scénarios présente dans le fichier-*result.txt* dont le format est le suivant :

```
ESA-PetriNet version 1.0 - Fichier Résultats -

RED : places appartenant à l'état redouté

Scénario 1 : (redouté)
  E  : { instances de tir de transitions internes ou appelantes et évenements i, e et f }
  E2 : { instances de tir de transitions appelées }
  A  : { liens directs liant des éléments de E et/ou de E2, avec le nom des atomes }
  B  : { liens indirects liant des éléments de E et/ou de E2 }
  C  : { liens d'appels de méthodes liant des éléments de E avec des éléments de E2 }
  Le : { jetons d'enrichissement }

Scénario 2 : (non-redouté)
  E  : { instances de tir de transitions internes ou appelantes et évenements i, e et f }
  E2 : { instances de tir de transitions appelées }
  A  : { liens directs liant des éléments de E et/ou de E2, avec le nom des atomes }
  B  : { liens indirects liant des éléments de E et/ou de E2 }
  C  : { liens d'appels de méthodes liant des éléments de E avec des éléments de E2 }
  Le : { jetons d'enrichissement }
[…]
```

Figure 7 : Format d'un fichier-result.txt

RED, ensemble des places appartenant à l'état redouté, est une liste de *nomObj*/*nomPlace* séparés par des espaces.

Un scénario peut être **redouté** ou **non-redouté** (suivant les options sélectionnés – voir section 2.2).

E est une liste de *nomObj*/*nomEvt* séparés par des espaces, où *nomEvt* est :

- soit un nom de transition interne ou appelante augmenté par .*numéroOccurence* (par exemple : **EV1/def1.1**),
- soit un événement initial (i1, i2,…), un événement d'enrichissement (e1, e2,…) ou un événement final (f1, f2,…) (par exemple : **EV1/f2**).

E2 est une liste de *nomObj*/*nomEvt* séparés par des espaces, où *nomEvt* est un nom de transition appelée.

A est une liste de (*nomObj1*/*nomEvt1*,*nomObj2*/*nomEvt2*)*nomPlace* séparés par des espaces, où *nomPlace* représente l'atome créé par *nomObj1*/*nomEvt1* et consommé par *nomObj2*/*nomEvt2*.

B et **C** sont des listes de (*nomObj1*/*nomEvt1*,*nomObj2*/*nomEvt2*) séparés par des espaces.

Le est une liste de (*nomObj*/e_i,*nomPlace*) séparés par des espaces, où *nomPlace* correspond à l'atome créé par l'événement d'enrichissement e_i dans le RdP de l'objet *nomObj*.

Note : si aucun fichier-*result.txt* n'est renseigné à travers l'interface lorsque la recherche des scénarios est lancée, un fichier-*result.txt* est automatiquement créé dans le même répertoire que le premier fichier de la liste des fichiers-*struct.txt* et le champ de l'interface correspondant est automatiquement complété.

2.7 FICHIER CONTRAINTES

L'utilisation des fichiers de contraintes n'a pas été implémentée dans la version 1.0 de ESA PetriNet. Il s'agirait de vérifier des propriétés temporelles sur l'ensemble des scénarios générés, ce qui signifie que l'extraction des scénarios devrait se faire avant cette vérification et que, par conséquent, cette vérification n'est pas utile pour la génération des scénarios redoutés.

2.8 ZONE DES GRAPHES DE PRECEDENCES

Les graphes de précédences s'affichent dans la zone 4 de la figure 1 (section 2.1) après avoir cliqué sur le bouton Graphe de Précédence, et ceci d'après le fichier-*result.txt* renseigné dans l'interface. Il est possible d'ajouter des commentaires aux graphes de précédence dans des cadres présents à gauche de chaque graphe, comme le montre la figure 8.

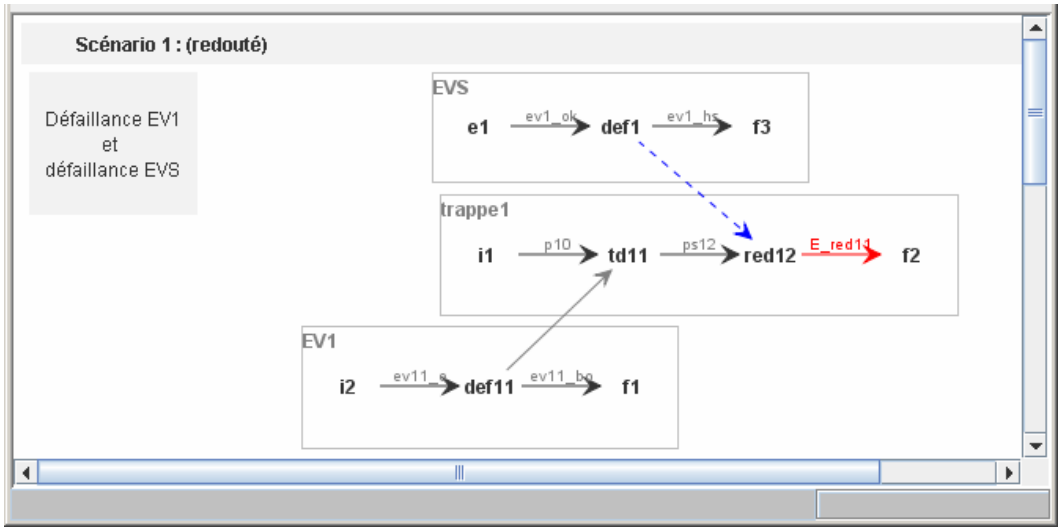


Figure 8 : Exemple de graphe de précédence commenté

Comme l'illustre le graphe ci-dessus, les événements sont regroupés par objet dans des rectangles déplaçables en blocs. Les flèches noires forment les liens directs, les bleues en pointillés représentent les liens indirects et les grises montrent les appels de méthodes entre objets. Les atomes appartenant à l'état redouté et leurs liens entre événements sont en rouge.

2.9 FICHIERS PROJET

Un projet peut être sauvegardé à tout moment et son enregistrement est composé de 2 fichiers.

FICHIER-ESA.TXT

Le premier est le fichier-*esa.txt*. Il contient toutes les données renseignées dans les différents champs de l'interface (fichier(s)-*struct.txt*, fichier-*com.txt* et/ou fichier-*result.txt*). Sa structure, dans sa forme la plus évoluée, est donnée en figure 9.

```
ESA-PetriNet version 1.0

net et inv : chemin d'accès au 1er fichier-struct.txt
net et inv : chemin d'accès au 2ème fichier-struct.txt
[...]
net et inv : chemin d'accès au nème fichier-struct.txt

com : chemin d'accès au fichier-com.txt

res : chemin d'accès au fichier-res.txt

cont : chemin d'accès au fichier-cont.txt
```

Figure 9 : Format d'un fichier-esa.txt

Note : suivant l'emplacement choisi pour l'enregistrement du projet, les chemins d'accès enregistrés seront soit **absolus** (exemple : « **c:/projet/fichier-*struct.txt* »**), soit **relatifs** (exemple : « **.../fichier-*struct.txt*** ») :

- Un chemin sera **relatif** lorsque le fichier à référencer se trouve dans le même répertoire que le fichier-*esa.txt* du projet ou dans l'un de ses sous-répertoires.
- Si ce n'est pas le cas, le chemin sera **absolu**.

De cette manière, le répertoire entier contenant le fichier-*esa.txt* du projet peut être déplacé, tout en conservant des références valides. Par contre, si les fichiers qui sont référencés de manière absolue sont déplacés, le lien sera rompu.

FICHIER-ESA.TXT.GRAPH

Le deuxième fichier à être enregistré lors de la sauvegarde d'un projet est le fichier-*esa.txt.graph*. Il contient, s'ils existent, les graphes de précédence affichés ainsi que les éventuels commentaires associés à ces graphes.

Remarque : contrairement à tous les autres fichiers, celui-ci n'est pas lisible par un éditeur de texte.

2.10 LES MESSAGES D'ERREURS

Le tableau si dessous donne la liste de tous les messages d'erreurs qui peuvent apparaître dans la barre d'état de la fenêtre principale de ESA PetriNet. Chaque erreur y est expliquée et certaines actions pour pallier à ces erreurs sont fournies.

Message d'erreur	***Explication***	***Action/Solution***
Impossible de sauvegarder l'adresse de Tina	L'écrire du fichier qui doit stoker le chemin d'accès à l'exécutable de TINA n'est pas possible.	1. Vérifier qu'il n'y ait pas un fichier nommé *adresseTina.txt* interdit en écriture dans le même répertoire que l'exécutable de ESA PetriNet. 2. Vérifier que le répertoire où ce trouve l'exécutable de ESA PetriNet soit autorisé en écriture.
Impossible de trouver Tina ou le fichier *.ndr	Lorsqu'un fichier-*struct.txt* est sélectionné et que l'on cherche à ouvrir TINA depuis ESA PetriNet, ce message apparaît si l'exécutable de TINA ou le fichier*.ndr* du fichier-*struct.txt* ne sont pas trouvés.	1. Vérifier que le fichier*.ndr* et le fichier-*struct.txt* se trouvent dans le même répertoire. 2. Redéfinir le chemin d'accès à l'exécutable de TINA à l'aide du menu **Option** (voir section 2.2)
Impossible de trouver Tina	Lorsque aucun fichier-*struct.txt* n'est sélectionné et que l'on cherche à ouvrir TINA depuis ESA PetriNet, ce message apparaît si l'exécutable de TINA n'est pas trouvé.	Redéfinir le chemin d'accès à l'exécutable de TINA à l'aide du menu **Option** (voir section 2.2)
Impossible d'ajouter 2 fois le même fichier	Il n'est pas possible d'ajouter 2 fois le même fichier-struct.txt à la liste des fichiers définissant les objets RdP.	Si toute fois vous souhaité utilisé 2 fois le même objet RdP dans votre modèle, copier le fichier-*struct.txt* vers un autre nom de fichier et ajouter ce nouveau fichier à la liste.
Impossible de sérialiser les graphes	Une erreur c'est produite lors de l'enregistrement des graphes de précédence du projet à sauvegarder.	1. Recommencer la sauvegarde du projet. 2. Si l'action précédente ne résout pas le problème, régénérer les graphes de précédence avant de tenter une nouvelle sauvegarde du projet.
Impossible de sauvegarder le fichier dans le répertoire sélectionné	La sauvegarde du projet ne peut pas être réalisée dans le répertoire choisi.	1. Vérifier qu'aucun projet de même nom n'existe dans le répertoire choisi. 2. Vérifier que le répertoire est autorisé en écriture, sinon : choisir un autre répertoire.
Fichier non reconnu	Lors du chargement du projet, le logiciel ne reconnait pas un fichier projet ESA PetriNet version 1.0.	1. Vérifier que le fichier sélectionné est bien celui du projet. 2. Si l'erreur persiste, éditer le fichier du projet et vérifier son contenu ou recommencer le projet et faite une nouvelle sauvegarde.
Impossible de charger le fichier *-esa.txt.	Il est impossible de lire le projet sélectionné.	Vérifier que le fichier du projet est accessible en lecture.

3. LES ETAPES DE L'UTILISATION SUR UN EXEMPLE

3.1 PRESENTATION DE L'EXEMPLE

Dans cette partie du manuel, un tutorial illustre la démarche à suivre pour utiliser ESA PetriNet.

L'exemple traité se base sur une modélisation orientée objets d'un système de régulation du volume d'un réservoir. Le système est constitué d'un calculateur, d'une pompe, de deux électrovannes, d'un capteur de volume, d'un réservoir régulé et d'un réservoir de vidange (voir figure 10). Le réservoir régulé alimente des utilisateurs et le but du calculateur est de maintenir le volume entre deux valeurs prédéfinies : V_{min} et V_{max}. Pour ce faire, le calculateur dispose de l'information fournie par le capteur et commande l'électrovanne principale EV1. Si cette dernière tombe en panne, le calculateur peut encore agir sur le volume de liquide dans le réservoir par l'intermédiaire de l'électrovanne de secours (EVS), tant que celle-ci reste opérationnelle…

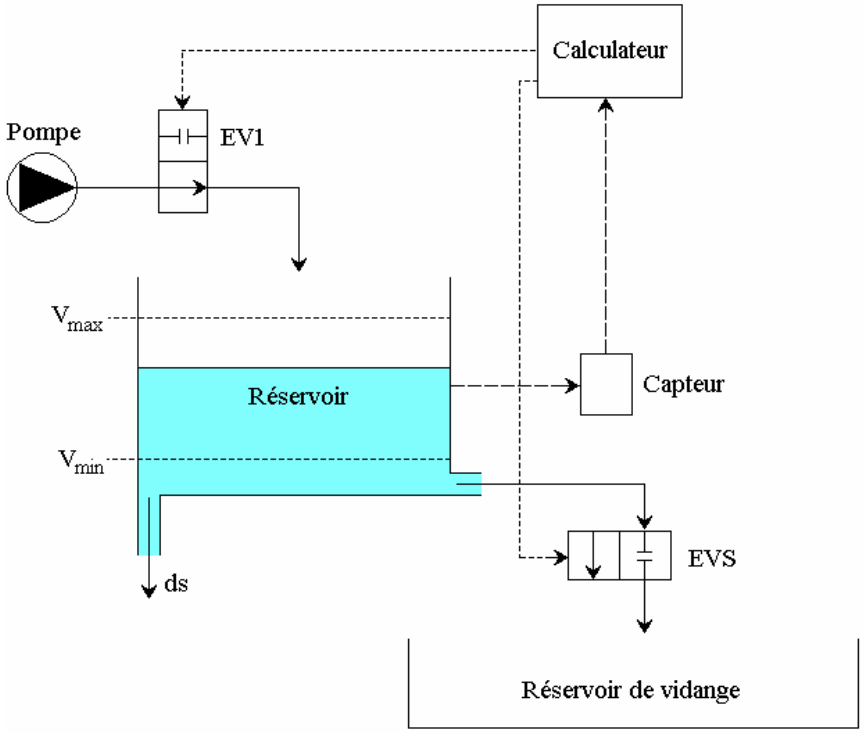


Figure 10 : Système de régulation d'un réservoir

Les différents réseaux de Petri utilisés pour modélise ce système sont donnés en figure 11, où, pour simplifier, seul trois objets sont considérés : le réservoir (principal), l'EV1 et l'EVS. Sur cette figure, les réseaux de Petri sont « formatés » pour ESA PetriNet et la présence d'un jeton dans la place E_red1 indique le choix de l'état redouté à analyser (il correspond au débordement du réservoir). Les communications entre objets sont aussi notifiées sur le schéma à travers 3 appels à méthode.

Remarque : les détails de modélisation de ce système sont expliqués dans la thèse de S. Khalfaoui soutenue le 26 septembre 2003 – (rapport LAAS n° 03574).

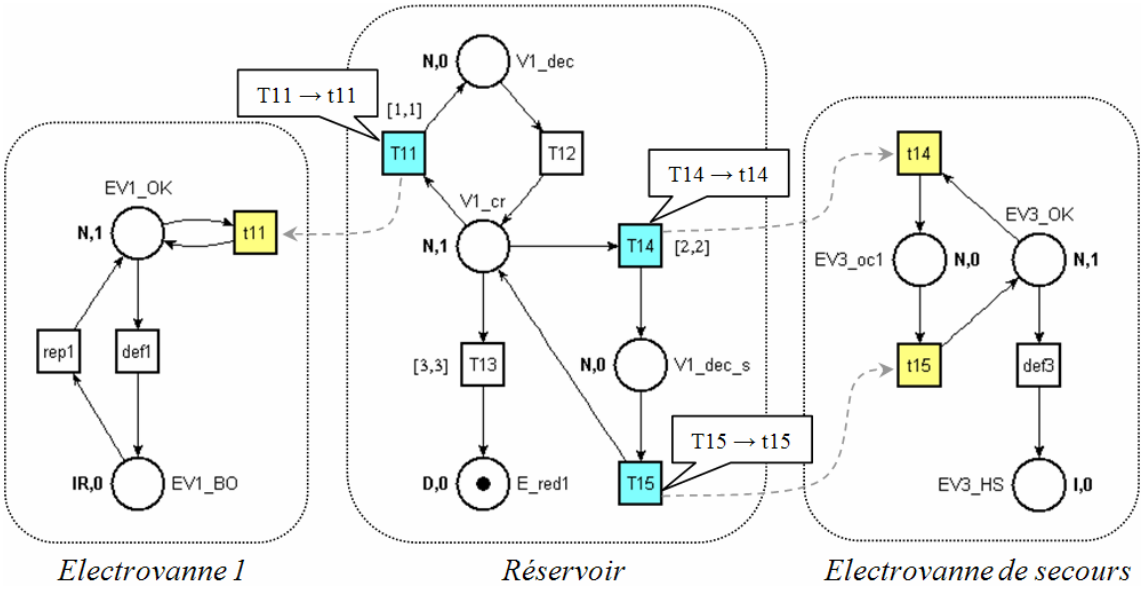


Figure 11 : Modèle RdP orienté objets du système de régulation d'un réservoir

3.2 EDITER LES RDP AVEC TINA

La première étape consiste à reproduire les 3 RdP « bien formatés » ci-dessus sous **TINA** dans des fichiers séparés. Ainsi, vous obtiendrez trois fichiers.*ndr* comme le montre la figure 12.

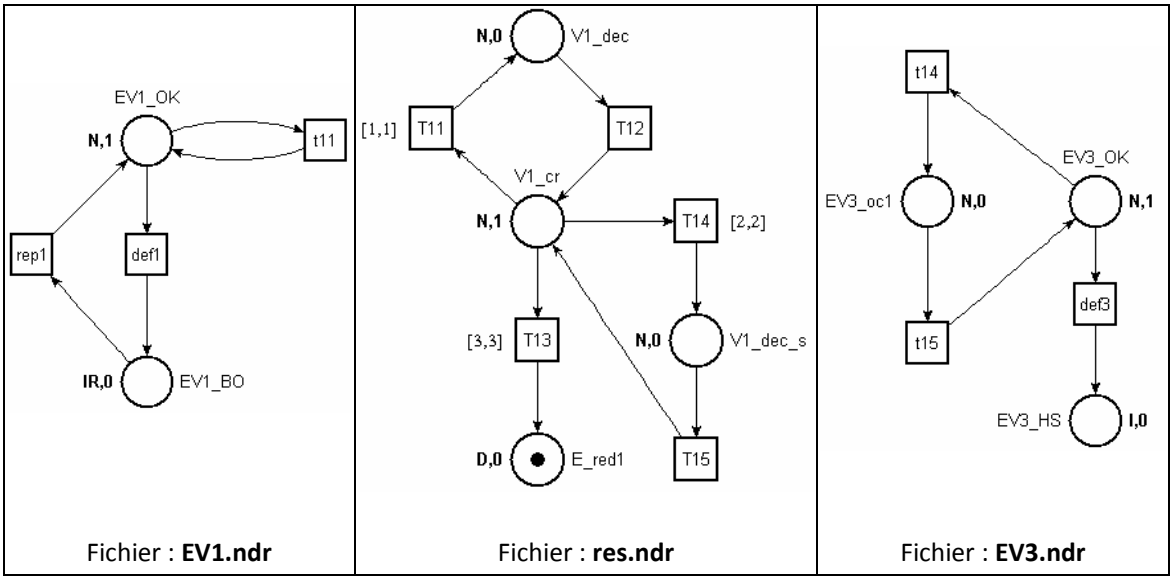
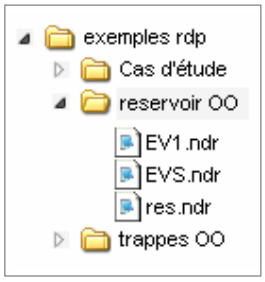


Figure 12 : Les 3 fichiers.ndr

3.3 ENREGISTRER LES FICHIERS.NDR

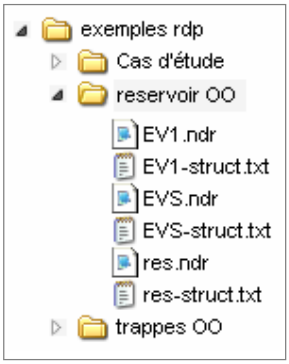


Pour un suivi aisé et une sauvegarde claire du projet, il est conseillé de créer un dossier spécifique pour chaque projet. Ici il s'agit de ***reservoir OO***. Les différents fichiers*.ndr* du projet sont alors enregistré dans ce répertoire. Pour un projet important, il peut être intéressant de regrouper les fichiers*.ndr* dans des sous-répertoires (des « packages »), afin d'organiser l'ensemble des objets utilisés. Par exemple, nous aurions pu créer un sous-répertoire ***electrovanne*** qui contiendrait ***EV1.ndr*** et ***EVS.ndr***.

3.4 GENERER LES FICHIERS-STRUCT.TXT

Ensuite, toujours sous l'outil **TINA**, il s'agit d'effectuer une analyse structurelle pour chaque objet RdP et de sauvegarder tous les fichiers-*struct.txt* issus de ces analyses **à coté de leurs fichiers*.ndr* respectifs**.

Pour ce faire, il faut, pour chaque fichier*.ndr* ouvert dans TINA, cliquer sur le menu ***tools***, puis sélectionner ***structural analysis***, laisser les options par défaut et cliquer sur ***OK***. A l'issue, le contenu (du texte) du futur fichier-*struct.ndr* apparaît dans une nouvelle fenêtre. Pour en faire la sauvegarde, il suffit de vouloir fermer la fenêtre (bouton **X** en haut à droite).



3.5 COMPLETER L'INTERFACE

Maintenant, vous pouvez fermer TINA et lancer ESA PetriNet. Vous devez alors renseigner les 3 fichiers-*struct.txt* dans l'interface (voir section 2.1). Pour cela, cliquez sur le bouton en forme de dossier à droite de la zone 2 et en haut, dont le message est **Ajouter un fichier Struct**. Sélectionnez le fichier à ajouter et cliquez sur **Ouvrir** pour effectivement l'ajouter à la liste des fichiers-*struct.txt* de l'interface. Répétez l'opération autant de fois qu'il y a de fichiers-*struct.txt*, c'est-à-dire 3 fois dans notre cas : pour *EV1-struct.txt*, pour *EVS-struct.txt* et pour *res-struct.txt* (voir figure 13).

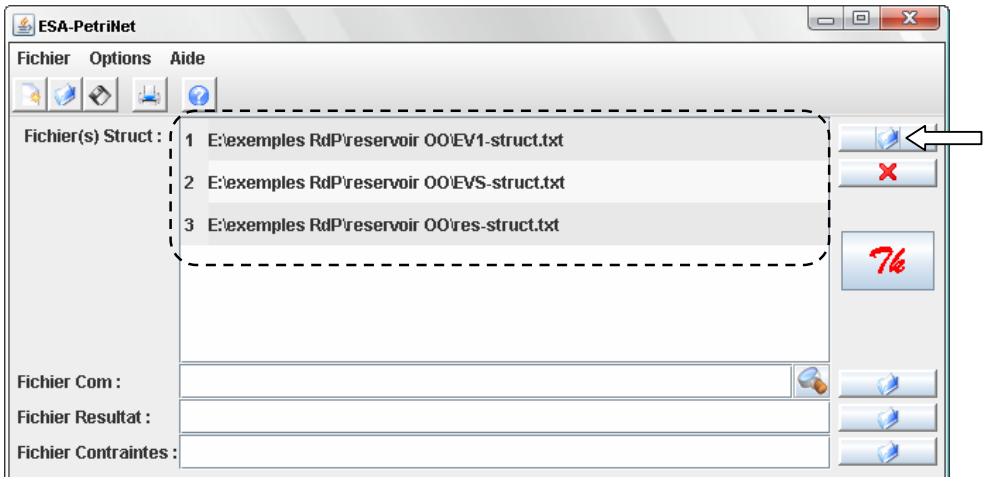


Figure 13 : Fichiers-struct.txt renseignés dans l'interface

3.6 CREER LE FICHIER-COM.TXT

Une fois seulement que tous les fichiers-*struct.txt* ont été renseignés dans l'interface, vous pouvez ouvrir l'éditeur de fichier de communications en cliquant sur la loupe (voir section 2.5). Il s'agit alors de définir les appels à méthodes que montre la figure 11.

Dans un premier temps, vous allez définir les trois transitions appelantes : **T11**, **T14** et **T15**, en les sélectionnant une à une dans la liste de gauche et en cliquant à chaque fois sur le bouton « **>>** » de la liste des transitions appelantes.

Ensuite, pour définir les transitions appelées, vous devez sélectionner une à une les transitions appelantes (par exemple **T11** dans un premier temps) et, pour chacune des ces transitions, sélectionner une à une les transitions appelées par la transition appelante courante dans la liste de gauche et cliquer à chaque fois sur le bouton « **>>** » de la liste des transitions appelées (pour **T11**, la transition appelée à ajouter (il n'y en a qu'une) est **t11** – voir figure 14).

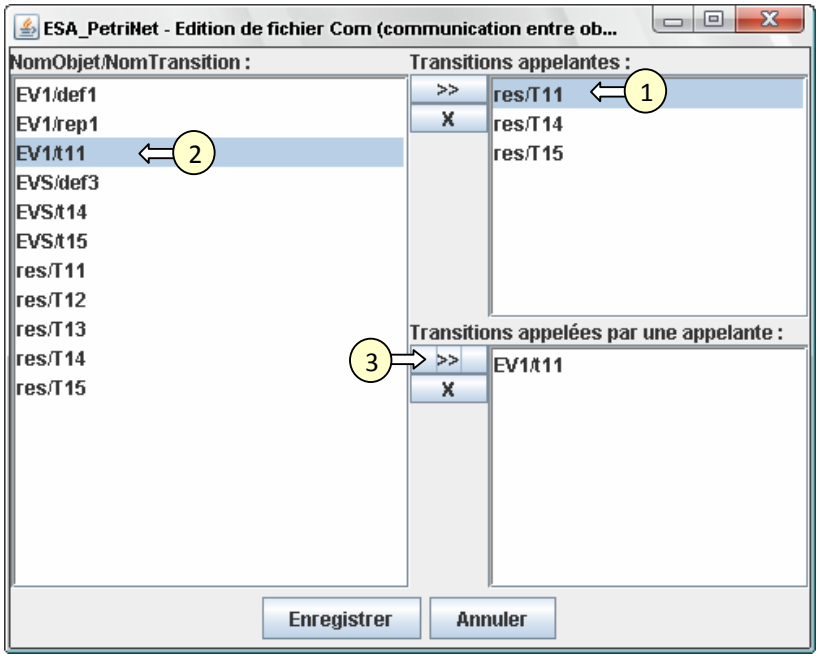
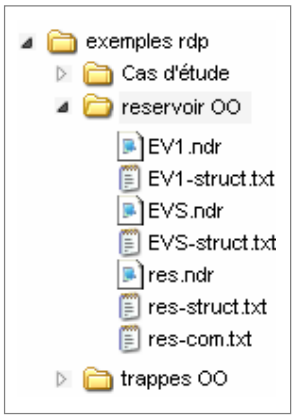


Figure 14 : Définition de t11 comme transition appelée par T11



Une fois que tous les appels de méthodes ont été définis, vous devez cliquer sur **Enregistrer**, choisir le répertoire du projet et nommer le fichier à créer : *res-com.txt* dans notre cas. Après avoir enregistré le fichier, l'éditeur de fichier de communications se ferme et le champ de l'interface relatif au fichier-*com.txt* se complète automatiquement.

Ci-dessous, la figure 15 présente le contenu du fichier *res-com.txt*.

```
Fichier Communications version 1.0

res/T11 : EV1/t11
res/T14 : EVS/t14
res/T15 : EVS/t15
```

Figure 15 : Fichier res-com.txt

3.7 GENERER LES SCENARIOS

Arrivé à cette étape, vous allez demander à ESA PetriNet de générer les scénarios. Par défaut, comme les options n'ont pas été changées, les scénarios générés seront les scénarios redoutés minimaux uniquement.

Pour cela, vous devez cliquer sur le bouton **Générer les scénarios**. Comme aucun fichier de résultat n'a été spécifié, un fichier par défaut est créé automatiquement : *esa-result.txt*.

Ensuite, cliquez sur **Graphe de Précédence** pour afficher les scénarios contenus dans ce fichier de résultat *esa-result.txt* sous la forme de graphes de précédence (voir figure 16).

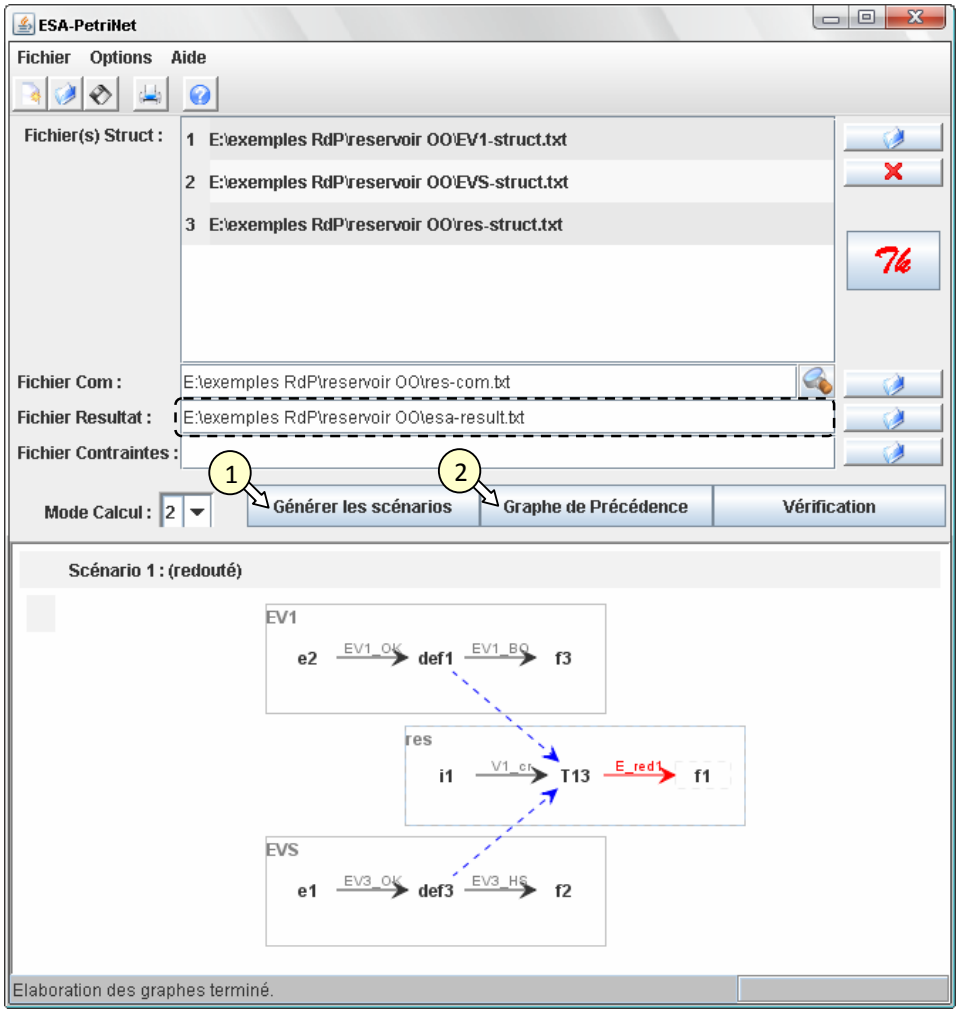


Figure 16 : Graphe de précédence généré

3.8 COMMENTER LES SCENARIOS

Une fois les scénarios générés et affichés, il est possible de commenter chaque scénario dans une zone grisée, à gauche de chaque graphe. Par exemple, dans notre cas nous pouvons ajouter le commentaire : « **défaillance de l'EV1 et défaillance de l'EVS** » (voir figure 17).

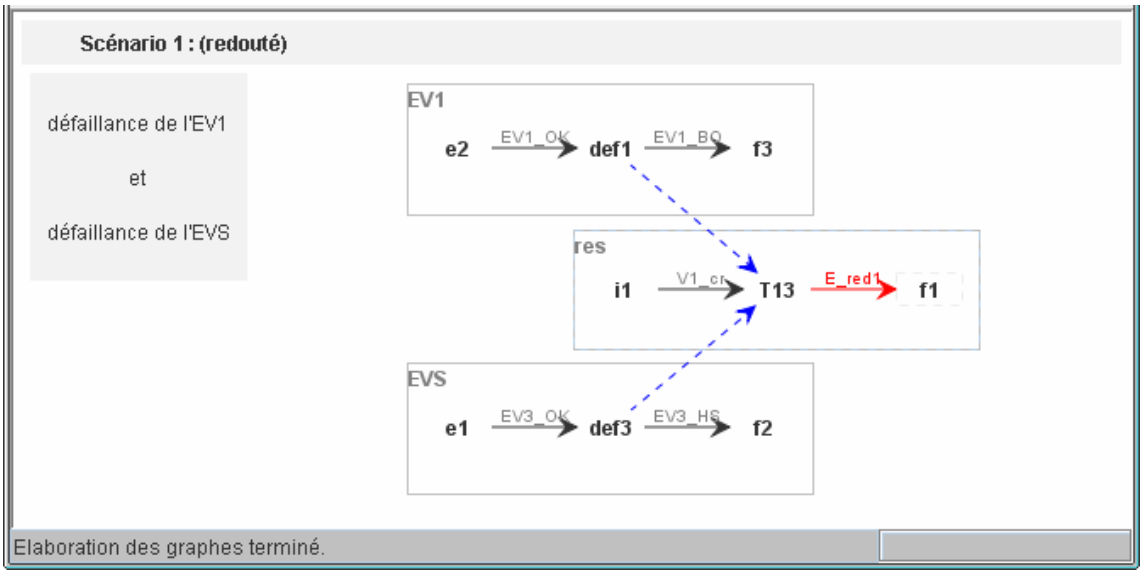
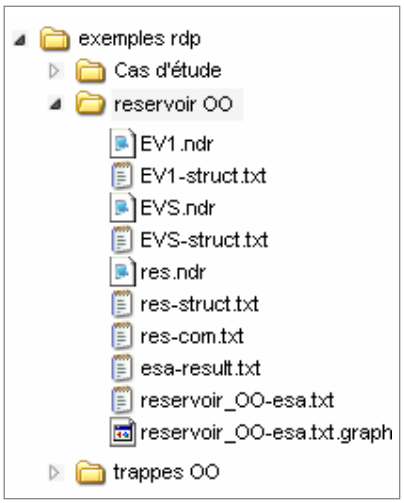


Figure 17 : Graphe de précédence commenté

3.9 SAUVEGARDER LE PROJET



Dans une dernière étape, nous allons sauvegarder le projet. De manière très intuitive, vous trouverez dans le menu **Fichier** : la commande **Enregistrer**. Cliquez sur cette commande, puis sélectionnez le répertoire du projet : **reservoir OO**, et nommé le projet, par exemple par : **reservoir_OO-esa.txt**.

2 nouveaux fichiers se sont alors créés dans le répertoire du projet : **reservoir_OO-esa.txt**, qui contient l'information des champs de l'interface, et **reservoir_OO-esa.txt.graph** qui est l'enregistrement du graphe de précédence affiché et de son commentaire.

Ainsi, il est possible de rouvrir le projet par la suite, ceci en cliquant sur la commande **Ouvrir** du menu **Fichier**, puis en sélectionnant le fichier **reservoir_OO-esa.txt**.

4. NOTE

Auteur : Hamid DEMMOU, Nabil SADOU et Romaric GUILLERM au LAAS-CNRS
7, avenue du Colonel Roche, 31077 TOULOUSE Cedex 4 - FRANCE